Live-Virtual-Constructive (LVC) Federation Engineering

Marine Corps M&S Management Office (MCMSMO)

Presented by Katherine L. Morse, PhD
JHU/APL
27 August 2015

Unclassified//U

Class Outline

- **Training goal and objectives**
- **Federation engineering standards**
- **Federation engineering issues**

Training Goal

- **Problem -** The possibility exists that many disparate and isolated LVC related efforts occurring within the Department of the Navy (DoN) may produce incompatible solutions (both amongst themselves and with other Joint solutions) which will:
  - Create duplicative capabilities,
  - Consume limited DoN resources, and
  - Introduce unnecessary technical and operational risks.
- **Solution** - Educate LVC users to make informed decisions based on knowledge of LVC:
  - Terminology
  - ***LVC Federation Engineering***
  - Technologies
  - Policy and stakeholders

Training Objectives

- Learn to make informed decisions based on knowledge of LVC
  - ELO 1: Demonstrate familiarity with LVC federation engineering processes and standards.
  - ELO 2: Identify federation engineering issues to address when applying and integrating LVC technologies.

Return on Investment (ROI) for Applying Federation Engineering Standards

- **Standards produced by internationally recognized standards development organizations (SDOs) receive input and review from subject matter experts (SMEs) in the field, ensuring comprehensiveness and quality.**
  - **Reducing errors that consume limited DoN resources, and introduce unnecessary technical and operational risks.**
- **Adopting such standards enables consistency of understanding and application across team members.**
  - **Reducing the probability of misunderstandings that introduce unnecessary technical risks.**
- **The standards applied in this class are developed and maintained by the Simulation Interoperability Standards Organization (SISO), the only SDO devoted solely to simulation standards.**
  - **Applying SISO standards improve the probability of creating interoperable rather than duplicative capabilities.**

Federation Engineering Standards

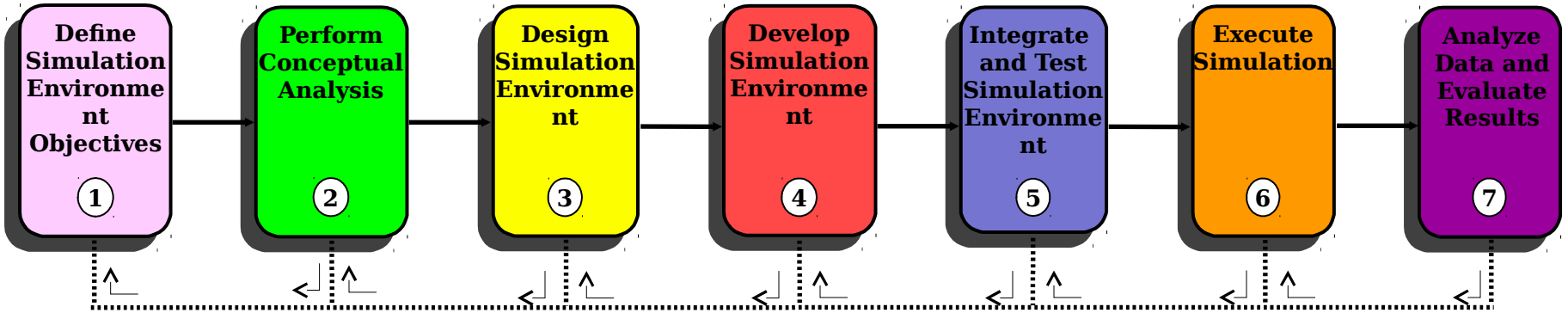
- **Distributed Simulation Engineering and Execution Process (DSEEP)**
  - **An architecture-neutral, high-level process framework into which the lower-level systems engineering practices native to any distributed simulation user can be easily integrated.**
  - **Applies to federation engineering in a single-architecture environment.**
  - ***DSEEP will be used to walk through the "how" of LVC.***
- **DSEEP Multi-Architecture Overlay (DMAO)**
  - **A recommended practice for applying the DSEEP to the development and execution of distributed simulation environments that include more than one distributed simulation architecture.**
  - ***DMAO will be used to identify typical major issues that arise in Department of Defense (DoD) LVC.***

Federation Engineering Standards

- **Federation Engineering Agreements Template (FEAT)**
  - **An XML schema designed for recording all federation agreements of use to federation developers and participants.**
  - **Serves as a format for recording agreements as well as a guide to identifying agreements necessary to complete and successful federation engineering.**
  - ***FEAT will be used to document all decisions made during LVC federation engineering.***

DSEEP Top Level View

- **The DSEEP was developed based on several, widely-adopted, authoritative systems engineering processes.**
- **These processes were adapted and extended to address engineering requirements unique to distributed simulations.**



DSEEP Activities

Step	(1) Define Simulation Environment Objectives	(2) Perform Conceptual Analysis	(3) Design Simulation Environment	(4) Develop Simulation Environment	(5) Integrate and Test Simulation Environment	(6) Execute Simulation	(7) Analyze Data and Evaluate Results
Activities	Identify User/Sponsor Needs Develop Objectives Conduct Initial Planning	Develop Scenario Develop Conceptual Model Develop Simulation Environment Requirements	Select Member Applications Design Simulation Environment Prepare Detailed Plan Design Member Applications	Develop Simulation Data Exchange Model Establish Simulation Environment Agreements Implement Member Application Designs Implement Simulation Environment Infrastructure	Plan Execution Integrate Simulation Environment Test Simulation Environment	Execute Simulation Prepare Simulation Environment Outputs	Analyze Data Evaluate and Feedback Results

Each major step of the DSEEP consists of multiple activities, each of which has multiple inputs, tasks, and outcomes that must be evaluated during federation engineering.

DMAO Overview

- **LVC environments are mixed architecture environments.**
- **DMAO extends the DSEEP by identifying issues unique to LVC development and defining best practices for addressing those issues.**
- **It's unlikely that a single mixed-architecture environment will have all of these issues, but all mixed-architecture environments will have some of them.**

Step	(1) Define Simulation Environment Objectives (2 issues)	(2) Perform Conceptual Analysis (2 issues)	(3) Design Simulation Environment (22 issues)	(4) Develop Simulation Environment (7 issues)	(5) Integrate and Test Simulation Environment (2 issues)	(6) Execute Simulation (1 issue)	(7) Analyze Data and Evaluate Results (1 issue)
Activities	Identify User/Sponsor Needs (no issues) Develop Objectives (no issues) Conduct Initial Planning (2 issues)	Develop Scenario (no issues) Develop Conceptual Model (no issues) Develop Simulation Environment Requirements (2 issues)	Select Member Applications (2 issues) Design Simulation Environment (14 issues) Prepare Detailed Plan (1 issue) Design Member Applications (5 issues)	Develop Simulation Data Exchange Model (2 issues) Establish Simulation Environment Agreements (1 issue) Implement Member Application Designs (2 issues) Implement Simulation Environment Infrastructure (2 issues)	Plan Execution (2 issues) Integrate Simulation Environment (1 issue) Test Simulation Environment (4 issues)	Execute Simulation (1 issue) Prepare Simulation Environment Outputs (no issues)	Analyze Data (no issues) Evaluate and Feedback Results (1 issue)

FEAT Overview

- **All LVC implementations need to establish federation agreements.**
- **Prior to the standardization of FEAT, there was no standard/convention for their structure or content.**
  - **Reducing the opportunity for reuse**
  - **Impeding comprehension by new federation participants.**
- **FEAT also serves as a checklist for identifying agreements that should be**

Just as the 5-paragraph order is a template to communicate a leader's military plans, the FEAT is a template that will document and communicate LVC federation engineering decisions.

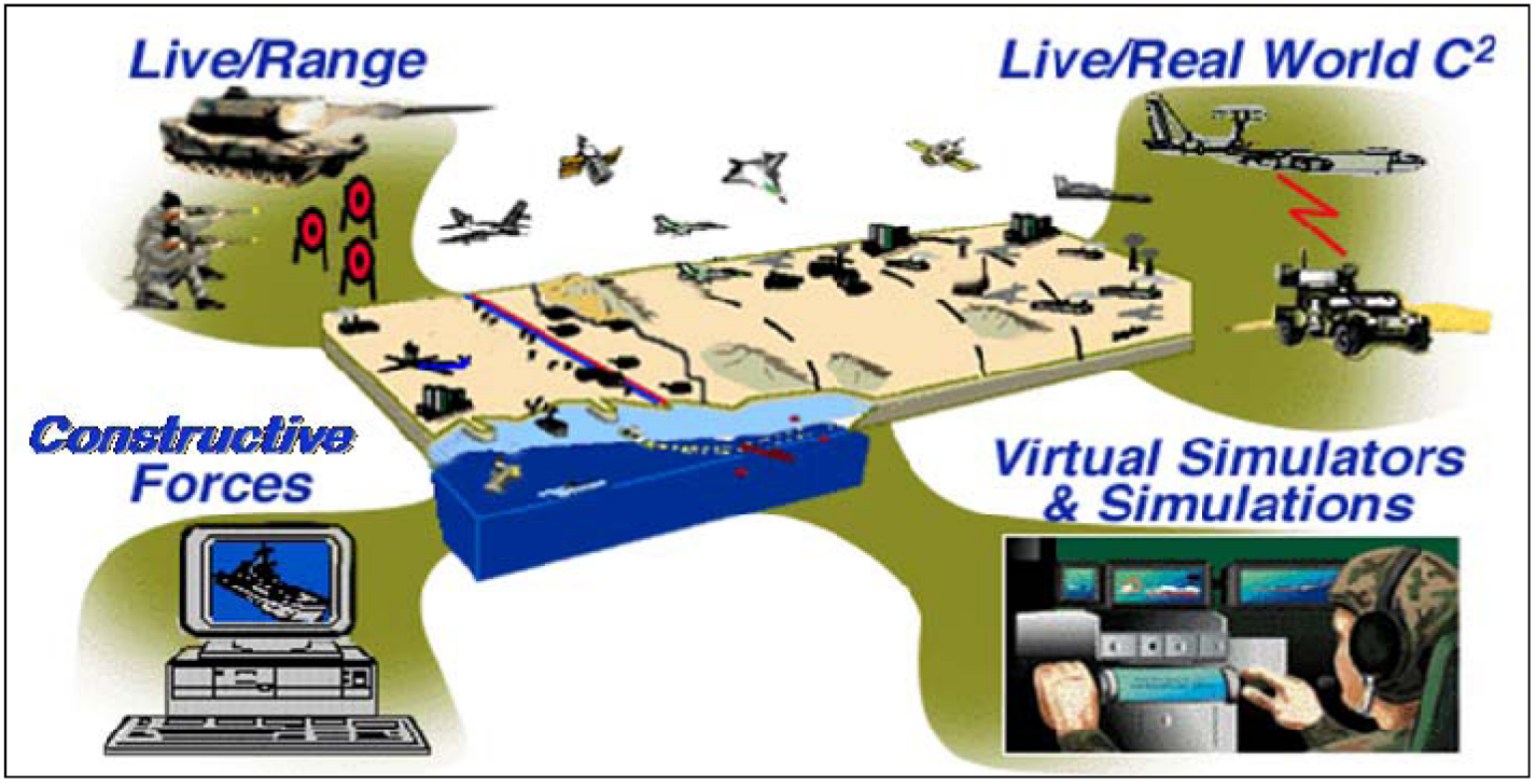
FEAT's "8-Paragraph" Categories

Category	Description
Design	Agreements about the basic purpose and design of the federation.
Execution	Technical and process agreements affecting execution of the federation.
Management	Systems/software engineering and project management agreements.
Data	Agreements about structure, values, and semantics of data to be exchanged during federation execution.
Infrastructure	Technical agreements about hardware, software, network protocols, and processes for implementing the infrastructure to support federation execution.
Modeling	Agreements to be implemented in the member applications that semantically affect the current execution of the federation.
Variances	Exceptions to the federation agreements deemed necessary during integration and testing.
Metadata	Information about the federation agreements document itself.

LVC Definitions

- **Live - Simulation involving real people operating real systems. Military training events using real equipment are live simulations. They are considered simulations because they are not conducted against a live enemy.**
- **Virtual - Simulation involving real people operating simulated systems. Virtual simulations inject human-in-the-loop in a central role by exercising motor control skills, e.g. flying an airplane and decision skills, e.g. committing fire control resources to action.**
- **Constructive - Models that involve simulated people operating simulated systems. Real people stimulate (make inputs) to such simulations, but are not involved in determining the outcomes, e.g. semi-automated force (SAF) simulations.**

Distributed LVC Environments



Requirements for LVC Environments

- The systems engineering process for LVC environments must begin with a clear operational requirement that is deconstructed into functional requirements which will set the criteria for technical requirements.
  - For example, a training and readiness event is deconstructed into several tasks and conditions that must be performed to a specific standard.
- Operational requirements are typically documented in DoD terminology that is unambiguous to the DoD audience.
- Technical requirements are typically documented in engineering terminology that is unambiguous to developers.
- Functional requirements

- Operational Requirement: The DoD relevant event that the LVC capability is targeted to simulate.
  - Used to make accreditation determinations.
- Functional Requirement: The tasks and conditions that the LVC capability must simulate to meet the operational requirement intent.
  - Used to make validation determinations.
- Technical Requirement: The software and hardware specifications that are required to be implemented to simulate the functional requirements.
  - Used to make verification determinations

LVC Federation Engineering Issues

- Each issue is presented on a single slide using this template.
- Each slide will include a brief description of the issue.
- Definitions will be provided for terms.

A reference block will direct the student to sections in the DSEEP, DMAO, and FEAT that describe the issue and potential solutions in detail.

A bumper sticker will alert the student to the risk of not addressing the issue.

DSEEP	DMAO	FEAT

Data Exchange Model (DEM)

- **For an LVC environment to operate properly, applications must exchange data at runtime.**
- **Different LVC architectures format data using different structures, e.g.**
  - **Distributed Interactive Simulation (DIS) Protocol Data Units (PDUs)**
  - **High Level Architecture (HLA) interactions and object class attribute updates**
- **The format may also differ between versions of a single standard.**
- **Applications may represent the same type of data differently, e.g.**
  - **Coordinate systems**
  - **Direction and speed**

- **DEM - A specification defining the information exchanged at runtime to achieve a given set of simulation objectives.**
  - **This includes class relationships, data structures, parameters, and other relevant information.**

Simulations with mismatched DEMs cannot communicate.

DSEEP	DMAO	FEAT
4.4.1	4.4.1.1.2	Data/DataExchangeModels

Data Storage Formats

- **Input databases that represent the same type of phenomena may be in very different formats.**
  - **Terrain**
  - **Order of battle / laydown**
- **Accurate, lossless conversion between formats may not be technically feasible.**
- **Conversion can be expensive and time consuming.**

- **Lossless - without loss of precision or accuracy; conversion of a datum from one format to another and back again produces the original value.**

Data conversion will negatively impact cost and schedule.

DSEEP	DMAO	FEAT
N/A	N/A	Data/supportingDatabases

Protocol / Architecture / Middleware

- There can be incompatibilities at every level of protocol/architecture/middleware.
  - Different architecture / protocols
    - HLA vs. DIS vs. Test and Training Enabling Architecture (TENA)
  - Different versions of the same architecture / protocol
    - DIS v6 vs. v7
    - DoD HLA 1.3 vs. Institute for Electrical and Electronics Engineering (IEEE) 1516-2000 vs. IEEE 1516-2010 vs. HLA "Run-time Infrastructures" (RTIs) not completely compliant with any authoritative standard
  - Different implementations (middleware) of the same version of a standard
    - IEEE 1516-2010 RTIs from different vendors
- Communicating between these requires some form of gateway.

- Middleware - ***Software*** that connects or integrates other software modules or components, typically providing a set of communications or interaction functions that may be invoked by the linked modules.
- Gateway - A ***device*** that connects two systems, especially if the systems use different protocols.

Inserting gateways to overcome incompatibilities introduces cost and latency, and potentially impacts schedule.

DSEEP	DMAO	FEAT
N/A	N/A	Infrastructure/middleware

Latency

- If applications or architectures run in real time, latency may introduce causal (fair fight) anomalies between applications.
  - An third-party observer application may observe the effects of a munitions fire before observing the munitions fire if there is less latency between the observer and the target than the observer and the shooter.
- Latency can be exacerbated by the use of gateways that translate between architectures:
  - Latency for the time required to translate the data from one protocol to another
  - Latency for an additional network message transmission if the gateway(s) are separate network nodes
  - Each additional gateway introduces additional latency.

- **Latency** – time delay between any two simulations, from submitting a message from the sending simulation to receiving this message by the recipient simulation.
- **Fair fight** - differences in simulations' performance characteristics have significantly less effect on the outcome of the conflict than actions taken by the simulations.

Latency can introduce fair fight anomalies.		
DSEEP	DMAO	FEAT
N/A	4.3.2.1.1	N/A

Interest Management

- **Interest management approaches vary widely by architecture.**
- **Approaches include filtering by:**
  - **Message class**
  - **Message content**
  - **Geographic region**
  - **Site / Host identification**
  - **Functions external to message content or class**
- **These approaches can be very different semantically and functionally.**
- **Where translation between approaches is technically feasible, it usually requires a gateway.**
- **Interest management decisions are based on the simulation domain and scenario.**
  - **Ensure that too much data isn't filtered out because failure to receive some data can cause**

- **Interest management – data filtering capabilities designed to limit network data transmissions so applications receive only the data in which they are interested .**

Required data may be filtered out before it reaches a simulation, resulting in inconsistent state

DSEEP	DMAO	FEAT
N/A	4.3.2.1.5	Modeling/ddm

Time Management

- Applications may or may not have the ability to control the rate at which they advance time and / or order events.
  - Mission command (MC) systems usually do not operate accurately with any rate other than real-time.
- Time advancement strategies differ.
  - Scaled as a factor of wall clock time
    - Real-time is a special case of scaled where the scale is 1-1.
  - Discrete event, i.e. controlled by message time stamps
    - This can be faster or slower than real-time.
    - Mechanisms exist for rewinding to an earlier time, but are complex and infrequently used.
  - Synchronization points defined at scenario-specific events, e.g. once initial scenario generation is completed by all simulations
  - See [5, 6] for a discussion of how different time advancements strategies may and may not interoperate.
  - See [6] for a discussion of the consequences of causality failure in real-time simulations.
- The format and mechanism for sharing time

- Time management – maintaining a common sense of time among all federates (applications), either based on real time clock or based on an event based clock.
- Real-time - Simulated time advances at the same rate as actual time (wall clock time.

Mismatched time management strategies can introduce causality (fair fight) errors

DSEEP	DMAO	FEAT
4.2.3.2	4.3.2.1.4	Execution/timeManagement

Heartbeat Translation

- DIS requires Entity State PDUs be sent for all entities at a standard-specified minimum frequency, even if the entities are completely static.
  - The standard value for HRT_BEAT_TIMER is 5 seconds.
- This can create unnecessary message traffic and overhead in non-DIS applications that send object updates only when an attribute value has changed.

- **Heartbeat** – a periodic entity state update message used by DIS.
- **Timeout** – The period of real-world time after which the data associated with a PDU type for a specific object is cleared from a receiving simulation's database if no new PDU is received.

Failure to generate heartbeats by non-DIS applications can cause unintended entity deletion in DIS applications as a result of timeout

DSEEP	DMAO	FEAT
N/A	4.3.2.1.7	N/A

Save and Restore

- **While some architectures provide the hooks for initiating save and restore of applications, individual applications are usually responsible for saving and restoring their own state.**
  - **HLA has *Initiate Federate Save* and *Initiate Federate Restore***
- **Save and restore signals must be coordinated across architectures, possibly through gateways.**
- **Saving the state of individual applications must be synchronized to ensure that the federation can be restored to a consistent state for a specified point in time.**

- **-- No definitions --**

Individual simulations may not restore or may restore to a state inconsistent with the rest of the federation

DSEEP	DMAO	FEAT
4.4.2	4.3.2.1.12	Execution/executionStates

Initialization Sequencing and Synchronization

- **Some applications have explicit or implicit requirements for initialization ordering.**
  - **They assume actions will be taken by other applications in the federation before they can proceed.**
  - **HLA requires that federates explicitly "join" a federation to enable federation management [see Monitoring and Control].**
- **Gateways or middleware linking multiple architectures may exacerbate this issue.**
- **Federations that connect to operational systems or instrumentation may need those systems or instrumentation to be switched on prior to**

- **-- No definitions --**

Failure to synchronize may prevent some simulations from entering the federation or to enter in an

DSEEP	DMAO	FEAT
4.4.2	4.4.2.1.3	Execution/executionStates

Incompatible Algorithmic Models

- **Physical phenomena may be modeled in simulations using different and incompatible algorithms, e.g.**
  - **Dead reckoning**
  - **Aggregation**
  - **Effects adjudication**
- **If simulations logically engaged in modeling the same phenomenon use incompatible algorithms, their results can vary.**
  - **Potentially, this can impact the federation's validity, particularly fair fight.**

- **Aggregation – ability to group entities while preserving the collective effects of entity behavior and interaction while grouped.**
- **Dead reckoning - simulations maintain a simplified representation of the positions of nearby entities and extrapolate their last reported positions until the next state update information arrives.**
- **Effects adjudication – estimating the effects of individual simulations' actions on other simulations.**

Incompatible models can introduce inconsistent and invalid simulation states

DSEEP	DMAO	FEAT
4.2.2, 4.4.2	4.4.3.1.1	Modeling/deadReckoning, Modeling/effectsAdjudication, Modeling/aggregation

Network Configuration

- **Incompatibilities may be introduced by network protocol configuration options preferred by different simulation architectures including varying data formats.**
  - **DIS prefers User Datagram Protocol (UDP).**
  - **HLA RTIs may require multicast.**
    - **Multicast is not a standard setting for DoD routers.**
- **This may require conversion of data formats to the most efficient protocol.**
  - **Which in turn may require introducing gateways with attendant cost, schedule, and technical risk.**

- **User Datagram Protocol – a connectionless, best effort, "unreliable" protocol.**
- **Multicast - one-to-many or many-to-many distribution group communication where information is addressed to a group of destination computers**

Incompatible network configurations may prevent simulations from communicating.

4.4.4	4.4.4.1.1	Infrastructure/networking, Infrastructure/secondaryCommChannels

Monitoring and Control

- **Monitoring**
  - **Confirm that applications continue to execute as expected**
  - **Observe the status and disposition of simulated entities**
- **Control**
  - **Explicitly directing applications to perform management operations:**
    - **Initializing**
    - **Pausing / resuming**
    - **Joining and leaving the simulation**
  - **Some simulations also have the hooks to have individual entities controlled externally, e.g. moving their position.**
- **Without hooks and applications for monitoring and control of the simulation environment, a single simulation application can hang or crash the entire environment**

- **-- No definitions --**

A single simulation can hang or crash a federation without monitoring and control.

DSEEP	DMAO	FEAT
4.2.3, 4.6.1	4.6.1.1.1	Execution/monitoring, Execution/executionStates, Execution/joinAndResign

Data Collection

- **Data for analysis and after action review (AAR) may be collected in multiple locations in a simulation architecture.**
  - **Data collected at multiple locations must be collected, combined, correlated, and ordered to produce a coherent and consistent picture of the execution.**
- **Different definitions, formats, and resolutions for location (TSPI or PLI) exacerbate this process because it's used to correlate records for individual entity collected from different locations in the simulation architecture.**
  - **If position data for an individual entity is collected at multiple locations in the architecture, a decision must be made about which location is authoritative.**
- **Data collection tools may be**

- **TSPI – Time-Space-Position Information.**
- **PLI – Position Location Information.**

Inconsistent or incomplete data collection can make accurate AAR difficult or impossible.

DSEEP	DMAO	FEAT
4.2.3.2, 4.5.3	4.6.1.1.2	Execution/dataLogging

Network Security / Information Assurance

- **Operating in a secure environment require:**
  - **Cross domain solution(s) (CDSs), e.g. Radiant Mercury, for multi-level security.**
  - **Permission to connect and operate.**
- **Cross domain solutions are typically expensive to purchase, configure, and accredit.**
  - **They must be accredited for every configuration / environment in which they're used.**
- **Permission to connect and operate is absolute; there are no workarounds.**
  - **Determine security requirements and POCs as early as possible.**
  - **Maintain documentation of all communications and tasks performed.**

- **Multi-level security – use of an IT system to process information at different security classification levels.**
- **Cross domain solution – a hardware / software capability to manually or automatically access or transfer information between two or more differing security domains; a solution to one type of multi-level security problem.**

Federations and simulations that fail IA requirements will not be allowed to operate.

DSEEP	DMAO	FEAT
4.3.4	4.4.4.1.1	Infrastructure/networking

Ground Truth

- Data sent on the network describing the state of entities may not be “ground truth,” i.e., it may not correctly reflect the true state of simulated entities.
- Individual simulations may intentionally degrade or corrupt that ground truth information in situations where a system or entity they are simulating would not have access to perfect and complete information.
  - The exact location of an aircraft whose performance specifications are classified may be “fuzzed.”
- Simulations and control applications need to have a clear understanding of which data is ground truth and which isn’t.
  - See Data Collection and Terrain

- **Ground truth – the actual facts of a situation, without errors introduced by sensors or human perception and judgment; the actual state of a simulated entity.**

Misinterpreting which data is ground truth can introduce inconsistent simulation states.

DSEEP	DMAO	FEAT
N/A	4.3.2.1.9	N/A

Ownership Transfer

- **The ability to transfer ownership of entity attributes among simulations is required to support the cooperative modeling of a given object instance across a federation, e.g.**
  - **In a WAN environment, ownership of a fast-moving aircraft may be transferred to a simulation that is interacting with the aircraft to ensure network latency doesn't impact fair fight.**
- **Different architectures / protocols implement ownership transfer differently.**
- **Where translation between approaches is technically feasible, it usually requires a gateway.**
  - **As previously discussed, gateways introduce latency.**

- **Ownership - the authority to simulate, modify, and delete the attributes of a simulated entity.**

Incompatible implementations of ownership transfer can result in simulated entities being orphaned or multiple, inconsistent versions being modeled.

DSEEP	DMAO	FEAT
N/A	4.2.3.1.2, 4.3.2.1.3	Modeling/controlTransfers

Terrain

- **Different simulation communities and simulation products use different representations of terrain, e.g.**
  - **Synthetic Environment Data Representation and Interchange Specification (SEDRIS).**
  - **OpenFlight**
  - **Digital Terrain Elevation Data (DTED).**
- **Correlation of entity location between simulations using different representations is a common problem because lossless translation between the representations is often not possible.**
  - **Sometimes leading to errors like floating tanks.**

- **Terrain - the vertical and horizontal dimension of land surface; in simulation it can also encompass other aspects of the physical environment including soil properties, vegetation, and human-made structures.**
- **Intervisibility - the relative ability to be seen under given conditions of distance, light, atmosphere, etc.:**

Simulated entity locations can be physically impossible or terrain-dependent phenomena, e.g. intervisibility, can be inconsistent.

DSEEP	DMAO	PLAT
4.2.1	4.2.1	Design/terrain

References

1. IEEE Std 1730-2010, Distributed Simulation Engineering and Execution Process (DSEEP)
2. IEEE Std 1730.1-2013, Distributed Simulation Engineering and Execution Process (DSEEP) Multi-Architecture Overlay (DMAO)
3. SISO-STD-012-2013, Federation Engineering Agreements Template (FEAT)
4. DoD M&S Glossary
5. IEEE Std 1516.1-2010, Modeling and Simulation (M&S) High Level Architecture (HLA) – Federate Interface Specification
6. Richard Fujimoto, Parallel and Distributed Simulation Systems, 2000
7. IEEE Std 1278.1-2012, Distributed Interactive Simulation – Application Protocols